Downloaded from orbit.dtu.dk on: May 05, 2023

DTU Library

Simultaneously Exploiting Two Formulations: An Exact Benders Decomposition Approach

Lusby, Richard Martin ; Røpke, Stefan; Gamst, Mette; Spoorendonk, Simon

Publication date:
2018

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Lusby, R. M. (Author), Røpke, S. (Author), Gamst, M. (Author), & Spoorendonk, S. (Author). (2018). Simultaneously Exploiting Two Formulations: An Exact Benders Decomposition Approach. Sound/Visual production (digital)

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Simultaneously Exploiting Two Formulations: An Exact Benders Decomposition Approach

Richard M. Lusby   Stefan Røpke   Mette Gamst   Simon Spoorendonk

DTU Management Enginnering

The Technical University of Denmark

EURO 2018

Agenda

1. Combining Two MIP Formulations
    - Reformulations and Relaxations
    - Benders Decomposition
2. Application One
    - The Bin Packing Problem
3. Application Two
    - The Split Delivery Vehicle Routing Problem (SDVRP)
4. Summary

Motivation

- Multiple ways to model a problem using mixed integer linear programming
- Each formulation may have its own advantages and disadvantages
  - $\rightarrow$ Branching in one is easier than the other
  - $\rightarrow$ One provides a tighter relaxation than the other
- Can we couple two separate formulations together to exploit the benefits of each?
- Solve the resulting formulation using Benders Decomposition
- We consider two different cases:
  - $\rightarrow$ One formulation is a reformulation of the other
  - $\rightarrow$ One formulation is a relaxation of the other

Methodology (1)

$$(\mathcal{P}1): \texttt{min} \;\; c^T \boldsymbol{x}$$
$$s.t. \quad A_1 \boldsymbol{x} \geq b_1$$
$$\boldsymbol{x} \geq 0$$
$$\boldsymbol{x} \in \mathbb{Z}^{n_1}$$

$$(\mathcal{P}2): \texttt{min} \;\; f^T \boldsymbol{y}$$
$$s.t. \quad A_2 \boldsymbol{y} \geq b_2$$
$$\boldsymbol{y} \geq 0$$
$$\boldsymbol{y} \in \mathbb{Z}^{n_2}$$

$$(\mathcal{P}3): \texttt{min} \;\; f^T \boldsymbol{y},$$
$$s.t. \quad A_1 \boldsymbol{x} \geq b_1$$
$$A_2 \boldsymbol{y} \geq b_2$$
$$D\boldsymbol{x} = W\boldsymbol{y}$$
$$\boldsymbol{x}, \boldsymbol{y} \geq 0$$
$$\boldsymbol{x} \in \mathbb{Z}^{n_1}$$
$$\boldsymbol{y} \in \mathbb{Z}^{n_2}$$

Methodology (1)

$$(\mathcal{P}1): \texttt{min} \quad c^T\boldsymbol{x} \qquad s.t. \quad A_1\boldsymbol{x} \geq b_1, \quad \boldsymbol{x} \geq 0, \quad \boldsymbol{x} \in \mathbb{Z}^{n_1}$$

$$(\mathcal{P}2): \texttt{min} \quad f^T\boldsymbol{y} \qquad s.t. \quad A_2\boldsymbol{y} \geq b_2, \quad \boldsymbol{y} \geq 0, \quad \boldsymbol{y} \in \mathbb{Z}^{n_2}$$

$$\begin{aligned}(\mathcal{P}3): \texttt{min} \quad & f^T\boldsymbol{y}, \\ s.t. \quad & A_1\boldsymbol{x} \geq b_1 \\ & A_2\boldsymbol{y} \geq b_2 \\ & D\boldsymbol{x} = W\boldsymbol{y} \\ & \boldsymbol{x}, \boldsymbol{y} \geq 0 \\ & \boldsymbol{x} \in \mathbb{Z}^{n_1} \\ & \boldsymbol{y} \in \mathbb{Z}^{n_2}\end{aligned}$$

Methodology (1)

$$
\begin{aligned}
(\mathcal{P}1):\ \texttt{min}\ \ & c^T \boldsymbol{x} \\
s.t.\quad & A_1 \boldsymbol{x} \geq b_1 \\
& \boldsymbol{x} \geq 0 \\
& \boldsymbol{x} \in \mathbb{Z}^{n_1}
\end{aligned}
$$

$$
\begin{aligned}
(\mathcal{P}2):\ \texttt{min}\ \ & f^T \boldsymbol{y} \\
s.t.\quad & A_2 \boldsymbol{y} \geq b_2 \\
& \boldsymbol{y} \geq 0 \\
& \boldsymbol{y} \in \mathbb{Z}^{n_2}
\end{aligned}
$$

$$
\begin{aligned}
(\mathcal{P}3):\ \texttt{min}\ \ & f^T \boldsymbol{y}, \\
s.t.\quad & A_1 \boldsymbol{x} \geq b_1 \\
& A_2 \boldsymbol{y} \geq b_2 \\
& D\boldsymbol{x} = W\boldsymbol{y} \\
& \boldsymbol{x}, \boldsymbol{y} \geq 0 \\
& \boldsymbol{x} \in \mathbb{Z}^{n_1} \\
& \boldsymbol{y} \in \mathbb{Z}^{n_2}
\end{aligned}
$$

Methodology (2)
Branch-and-Benders-Cut

- Benders Decomposition is viewed as a Branch-and-Cut Algorithm

$$
\begin{aligned}
(\mathcal{MP}):\texttt{min}\ \ & f^T\boldsymbol{y} \\
s.t.\quad & A_2\boldsymbol{y} \geq b_2 \\
& \boldsymbol{y} \geq 0 \\
& \boldsymbol{y} \in \mathbb{R}^{n_2}
\end{aligned}
$$

$$
\begin{aligned}
(\mathcal{SP}):\texttt{min}\ \ & 0^T\boldsymbol{x} \\
s.t.\quad & A_1\boldsymbol{x} \geq b_1 \\
& D\boldsymbol{x} = W\bar{y} \\
& \boldsymbol{x} \geq 0 \\
& \boldsymbol{x} \in \mathbb{R}^{n_1}
\end{aligned}
$$

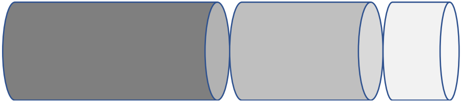
- Feasibility cuts are generated at each node of the Branch-and-Bound tree

$$u_1^T b_1 + u_2^T W\boldsymbol{y} \leq 0$$

- Branching can be implemented in either problem
- Branching on the $\boldsymbol{x}$ variables could be appealing
  - $\rightarrow$ Implications on $(\mathcal{MP})$ enforced through feasibility cuts

Application One: Bin Packing Problem
Reformulation

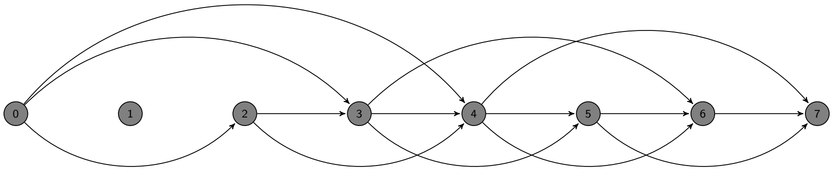
- The *Cutting Stock Problem* is a well known problem in the OR literature
- Involves determining the minimum number of rolls (or stocks) of a given length that must be cut in order to meet the demand for a set of items, of shorter, specified lengths.



- The *Bin Packing Problem* is a special case of this where the cost of using a given stock is equal to one.
- Various formulations exist
- Simultaneously consider an *arc-flow formulation* and a *path based formulation*

Formulations

The network - Carvalho (1999)



Arc flow formulation

$$(\mathcal{BP}1) \quad \texttt{min } z$$

$$\sum_{(j,i)\in A} \boldsymbol{x}_{j,i} - \sum_{(i,j)\in A} \boldsymbol{x}_{i,j} = \begin{cases} z & j = 0 \\ 0 & j = 1, \ldots, W-1 \\ -z & j = W \end{cases} \quad \forall j \in N$$

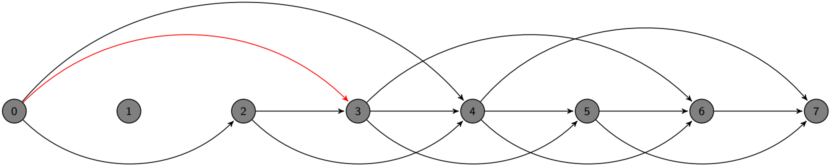
$$\sum_{(h,h+w_i)\in A} \boldsymbol{x}_{h,h+w_i} \geq b_i \quad \forall i \in I$$

$$\boldsymbol{x}_{ij} \in \mathbb{Z}^+ \quad \forall (i,j) \in A$$

$$z \in \mathbb{Z}^+$$

Formulations

The network - Carvalho (1999)



Arc flow formulation

$$(\mathcal{BP}1) \quad \texttt{min } z$$

$$\sum_{(j,i)\in A} \boldsymbol{x}_{j,i} - \sum_{(i,j)\in A} \boldsymbol{x}_{i,j} = \begin{cases} z & j = 0 \\ 0 & j = 1,\dots,W-1 \\ -z & j = W \end{cases} \quad \forall j \in N$$

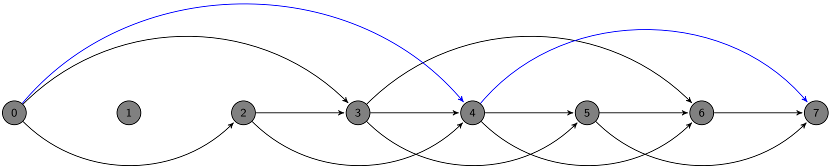
$$\sum_{(h,h+w_i)\in A} \boldsymbol{x}_{h,h+w_i} \geq b_i \quad \forall i \in I$$

$$\boldsymbol{x}_{ij} \in \mathbb{Z}^+ \quad \forall (i,j) \in A$$

$$z \in \mathbb{Z}^+$$

Formulations

The network - Carvalho (1999)



Pattern based

$$(\mathcal{BP}2) \quad \texttt{min} \sum_{p \in P} \boldsymbol{\lambda}_p$$

$$\sum_{p \in P} a_{ip} \boldsymbol{\lambda}_p \geq b_i \ \forall i \in I$$

$$\boldsymbol{\lambda}_p \in \mathbb{Z}^+ \quad \forall p \in P$$

Methodology

- Combine both formulations and augment with a linking constraint

$$
\begin{aligned}
(\mathcal{BP}3) \quad & \texttt{minimize} \sum_{p \in P} \boldsymbol{\lambda}_p, \\
& s.t. \quad \sum_{p \in P} a_{ip} \boldsymbol{\lambda}_p \geq b_i && \forall i \in I \\
& \sum_{p \in P} a_{ip} \boldsymbol{\lambda}_p = \sum_{(h, h+w_i) \in A} \boldsymbol{x}_{h,h+w_i} && \forall i \in I \\
& \sum_{(j,i) \in A} \boldsymbol{x}_{ji} - \sum_{(i,j) \in A} \boldsymbol{x}_{i,j} = \begin{cases} \sum_{p \in P} \boldsymbol{\lambda}_p & j = 0 \\ 0 & j = 1, \dots, W-1 \\ -\sum_{p \in P} \boldsymbol{\lambda}_p & j = W \end{cases} && \forall j \in N \\
& \boldsymbol{x}_{ij} \in \mathbb{Z}^+ && \forall (i,j) \in A \\
& \boldsymbol{\lambda}_p \in \mathbb{Z}^+ && \forall p \in P
\end{aligned}
$$

- Solve using Benders Branch-and-Cut
- Compare with the Branch-Price-and-Cut algorithm of Alves & Carvalho (2008)

Methodology

- Combine both formulations and augment with a linking constraint

$$
\begin{aligned}
(\mathcal{BP}3) \quad \texttt{minimize} \quad & \sum_{p \in P} \boldsymbol{\lambda}_p, \\
s.t. \quad & \sum_{p \in P} a_{ip} \boldsymbol{\lambda}_p \geq b_i && \forall i \in I \\
& \sum_{p \in P} a_{ip} \boldsymbol{\lambda}_p = \sum_{(h,h+w_i) \in A} \boldsymbol{x}_{h,h+w_i} && \forall i \in I \\
\sum_{(j,i) \in A} \boldsymbol{x}_{ji} - \sum_{(i,j) \in A} \boldsymbol{x}_{i,j} = & \begin{cases} \sum_{p \in P} \boldsymbol{\lambda}_p & j = 0 \\ 0 & j = 1, \dots, W-1 \\ -\sum_{p \in P} \boldsymbol{\lambda}_p & j = W \end{cases} && \forall j \in N \\
& \boldsymbol{x}_{ij} \in \mathbb{Z}^+ && \forall (i,j) \in A \\
& \boldsymbol{\lambda}_p \in \mathbb{Z}^+ && \forall p \in P
\end{aligned}
$$

- Solve using Benders Branch-and-Cut
- Compare with the Branch-Price-and-Cut algorithm of Alves & Carvalho (2008)

Methodology

- Combine both formulations and augment with a linking constraint

$$
\begin{aligned}
(\mathcal{BP}3) \quad \texttt{minimize} \quad & \sum_{p \in P} \boldsymbol{\lambda}_p, \\
s.t. \quad & \sum_{p \in P} a_{ip} \boldsymbol{\lambda}_p \geq b_i && \forall i \in I \\
& \sum_{p \in P} a_{ip} \boldsymbol{\lambda}_p = \sum_{(h,h+w_i) \in A} \boldsymbol{x}_{h,h+w_i} && \forall i \in I \\
\sum_{(j,i) \in A} \boldsymbol{x}_{ji} - \sum_{(i,j) \in A} \boldsymbol{x}_{i,j} = & \begin{cases} \sum_{p \in P} \boldsymbol{\lambda}_p & j = 0 \\ 0 & j = 1, \ldots, W-1 \\ -\sum_{p \in P} \boldsymbol{\lambda}_p & j = W \end{cases} && \forall j \in N \\
& \boldsymbol{x}_{ij} \in \mathbb{R}^+ && \forall (i,j) \in A \\
& \boldsymbol{\lambda}_p \in \mathbb{R}^+ && \forall p \in P
\end{aligned}
$$

- Solve using Benders Branch-and-Cut
- Compare with the Branch-Price-and-Cut algorithm of Alves & Carvalho (2008)

Instances

Category[1]	$\lvert I\rvert$	W	Instances
bpp1	120	150	8
bpp2	250	150	14
bpp3	500	150	11
bpp4	1000	150	11
bpp5	60	100	7
bpp6	120	100	20
bpp7	249	100	20
bpp8	501	100	20
Total			111

Algorithm coded in C++, uses COIN-BCP, Cplex 12.6 & Gurobi 7.5 for LPs

Max run time of 3 hours, Intel Xeon 2.6GHz processor, Linux

[1] https://www.math.u-bordeaux.fr/~fvanderb/publications.html

Results

	Alves & Carvalho (2008)			Benders Approach			
Class	Nodes	Price (s)	Solve (s)	Nodes	Cuts	Price (s)	Solve(s)
bpp1	73.00	0.62	4.04	24.34	0.00	0.49	1.60
bpp2	60.27	1.00	6.15	20.07	0.00	0.54	1.77
bpp3	200.50	7.48	88.56	48.82	0.00	0.49	4.56
bpp4	399.91	13.10	143.86	150.00	0.00	1.91	13.22
bpp5	26.00	1.75	3.08	29.20	0.00	0.29	5.44
bpp6	111.12	15.63	41.20	192.90	1.25	1.26	50.56
bpp7	423.60	187.88	933.07	1058.90	9.00	6.43	505.59
bpp8	778.80	769.35	3990.16	3873.50	67.75	30.56	3313.96

Results

	Alves & Carvalho (2008)			Benders Approach			
Class	Nodes	Price (s)	Solve (s)	Nodes	Cuts	Price (s)	Solve(s)
bpp1	73.00	0.62	4.04	24.34	0.00	0.49	1.60
bpp2	60.27	1.00	6.15	20.07	0.00	0.54	1.77
bpp3	200.50	7.48	88.56	48.82	0.00	0.49	4.56
bpp4	399.91	13.10	143.86	150.00	0.00	1.91	13.22
bpp5	26.00	1.75	3.08	29.20	0.00	0.29	5.44
bpp6	111.12	15.63	41.20	192.90	1.25	1.26	50.56
bpp7	423.60	187.88	933.07	1058.90	9.00	6.43	505.59
bpp8	778.80	769.35	3990.16	3873.50	67.75	30.56	3313.96

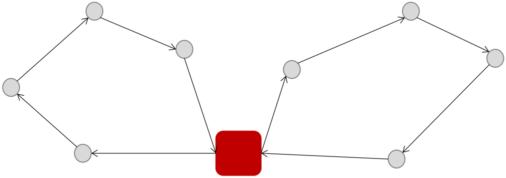
Comments

- Proposed methodology is competitive
- Benders Decomposition evaluates more nodes
- Feasibility cuts needed for some instances only

Application Two: SDVRP Relaxation

- Variant of the classical Capacitated Vehicle Routing Problem (CVRP)
- Given a set of *Customer demands*, *a depot*, and a fleet of *vehicles* with a certain capacity, partition the customers into vehicle tours that start and end at the depot, satisfy the vehicle capacities, and which minimize the transportation cost.

  Formulated on a Graph $G = (V, E)$



- In the SDVRP a customer can be visited by *multiple vehicles*
- Notoriously difficult problem
- Simultaneously consider an *exact formulation* and a *relaxation*

Formulations

- Use the exact three index formulation of Li et al. (2006)
  - $x_{ij}^k \in \{0,1\}$ determines with vehicle k travels between nodes i and j
  - $y_{ik} \in [0,1]$ states the portion of node i's demand serviced by vehicle k
  - $t_{ik} \geq 0$ is the time at which vehicle k visits node i
- Combine with the two index formulation of Archetti at al. (2014)
  - $x_{ij} \in \{0,1\}$ counts the number of vehicle's travelling between nodes i and j
  - $z_i \geq 0$ states the number of times node i is visited

Formulations

$$
\begin{aligned}
\min \quad & \sum_{i,j \in V} \sum_{k \in K} c_{ij} x_{ij}^k && \\
s.t. \quad & \sum_{j \in V} x_{0j}^k \leq 1 && \forall k \in K \\
& \sum_{j \in V} x_{ji}^k - \sum_{j \in V} x_{ij}^k = 0 && \forall i \in V \setminus \{0\}, \forall k \in K \\
& t_{ik} - t_{jk} + (|V| + 1) x_{ij}^k \leq |V| && \forall i, j \in V \setminus \{0\}, \forall k \in K \\
& \sum_{k \in K} y_{ik} = 1 && \forall i \in V \setminus \{0\} \\
& \sum_{i \in I} d_i y_{ik} \leq Q && \forall k \in K \\
& y_{ik} - \sum_{j \in V} x_{ji}^k \leq 0 && \forall i \in V \setminus \{0\}, \forall k \in K \\
& x_{ij}^k \in \{0, 1\} && \forall (ij) \in E, \forall k \in K \\
& 0 \leq y_{ik} \leq 1 && \forall i \in V, \forall k \in K \\
& t_{ik} \geq 0 && \forall i \in V, \forall k \in K
\end{aligned}
$$

Formulations

$$
\begin{aligned}
\min \quad & \sum_{i,j \in V} c_{ij} x_{ij} && \\
s.t. \quad & \sum_{i \in \delta(i)} x_{ij} = 2 z_i && \forall i \in V \\
& \sum_{(i,j) \in \delta(S)} x_{ij} \geq 2 \cdot \left\lceil \frac{\sum_{i \in S} d_i}{Q} \right\rceil && \forall S \subset N, |S| \geq 2 \\
& x_{ij} \in \{0, 1\} && \forall (i,j) \in E(N) \\
& k_N \leq x_{ij} \leq 2 \cdot |K| && \forall (i,j) \in E \setminus E(N) \\
& k_N \leq z_0 \leq |K| \text{ and integer} && \\
& 1 \leq z_i \leq |K| \text{ and integer} && \forall i \in N
\end{aligned}
$$

Methodology

- Couple both formulations together & augment the resulting model with

$$x_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i,j) \in E$$

- Apply Branch-and-Benders-Cut
- Use the two index formulation as the master problem
- Branch in the master problem
- Separate feasibility cuts on finding integer solutions
- Compare with the Branch-and-Cut Algorithm of Archetti et al. (2014)
    - Dynamically, and heuristically, separate capacity cuts, connectivity cuts

Results: Proposed Methodology

Tested on 25 instances from Belenguer et al. (2000) (Integer distances)

Instance	Ben.	Cap.	Con.	Subs	LP	IP	Nodes	Time (s)
eil22	0	47	0	1	375.00	375.00	1	0.11
eil23	0	20	1	1	569.00	569.00	1	0.06
eil30	1,216	3,055	1,430	2,021	508.00	510.00	4,679	109.71
eil33	0	163	0	1	833.50	835.00	3	1.46
eil51	0	5,221	81	12	514.10	521.00	237	17.35
S51D1	0	3,59	0	20	454.33	458.00	73	9.13
S76D1	0	10,750	80	16	586.16	592.00	399	63.66

Algorithm coded in C++, uses COIN-BCP, Cplex 12.6 & Gurobi 7.5 for LPs

Max run time of 3 hours, Intel Xeon 2.6GHz processor, Linux

Results: Proposed Methodology

Instance	Ben.	Cap.	Con.	Subs	LP^1	LP^2	IP	Nodes
S51D2	533	466,626	9,441	3,399	678.21	687.07	728.00	20,085
S51D3	560	302,259	5,566	2,761	905.78	914.31	N/A	14,079
S51D4	452	496,903	4,778	2,094	1,519.78	1,524.48	N/A	13,089
S51D5	560	539,575	6,118	2,601	1,279.09	1,287.24	N/A	17,118
S51D6	367	575,843	369	1,815	2,126.45	2,128.48	N/A	9,233
eilA76	399	318,952	9,790	2,519	778.10	785.04	N/A	14,432
eilB76	321	461,015	11,001	1,642	933.97	940.62	N/A	13,697
eilC76	162	1,467,051	32,989	1,236	703.78	709.56	780.00	39,742
eilD76	97	1,373,146	38,055	993	657.68	662.29	707.00	44,659
eilA101	24	914,453	17,926	298	788.19	793.13	872.00	22,541
eilB101	153	299,122	7,956	742	1,000.99	1,005.22	N/A	7,119
S76D2	320	480,082	10,431	1,627	1,016.56	1,022.03	N/A	12,525
S76D3	225	413,936	7,888	1,044	1,347.90	1,352.61	N/A	9,548
S76D4	124	340,259	3,774	743	2,005.18	2,007.16	N/A	7087
S101D1	58	867,619	8,785	961	702.77	708.46	726.00	24,211
S101D2	87	316,617	6,454	434	1,270.25	1,276.42	N/A	4,893
S101D3	77	369,606	4,668	348	1,753.37	1,755.25	N/A	4,211
S101D5	35	334,711	3,078	149	2,652.75	2,652.78	N/A	3,040

Results: Proposed Methodology

Instance	Ben.	Cap.	Con.	Subs	LP^1	LP^2	IP	Nodes
S51D2	533	466,626	9,441	3,399	678.21	687.07	728.00	20,085
S51D3	560	302,259	5,566	2,761	905.78	914.31	N/A	14,079
S51D4	452	496,903	4,778	2,094	1,519.78	1,524.48	N/A	13,089
S51D5	560	539,575	6,118	2,601	1,279.09	1,287.24	N/A	17,118
S51D6	367	575,843	369	1,815	2,126.45	2,128.48	N/A	9,233
eilA76	399	318,952	9,790	2,519	778.10	785.04	N/A	14,432
eilB76	321	461,015	11,001	1,642	933.97	940.62	N/A	13,697
eilC76	162	1,467,051	32,989	1,236	703.78	709.56	780.00	39,742
eilD76	97	1,373,146	38,055	993	657.68	662.29	707.00	44,659
eilA101	24	914,453	17,926	298	788.19	793.13	872.00	22,541
eilB101	153	299,122	7,956	742	1,000.99	1,005.22	N/A	7,119
S76D2	320	480,082	10,431	1,627	1,016.56	1,022.03	N/A	12,525
S76D3	225	413,936	7,888	1,044	1,347.90	1,352.61	N/A	9,548
S76D4	124	340,259	3,774	743	2,005.18	2,007.16	N/A	7087
S101D1	58	867,619	8,785	961	702.77	708.46	726.00	24,211
S101D2	87	316,617	6,454	434	1,270.25	1,276.42	N/A	4,893
S101D3	77	369,606	4,668	348	1,753.37	1,755.25	N/A	4,211
S101D5	35	334,711	3,078	149	2,652.75	2,652.78	N/A	3,040

Comments

- Struggles on larger instances
- Combine with an upper bound heuristic
- Feasibility cuts on fractional solutions

Summary

- Proposed framework couples two formulations to exploit both
- Tested its performance using a reformulation and a relaxation
- Shows promise
- More development and testing need

Thanks for your attention!